

STEMA 考试 C++试卷（2023 年 12 月）

一、 选择题

第一题

定义字符串 `string a = "Hello C++"`，下列选项可以获取到字符 'C' 的是（ ）。

- A、`a[7]`
- B、`a[6]`
- C、`a[5]`
- D、`a[4]`

第二题

下列选项中数值与其它项不同的是（ ）。

- A、 $(1234)_5$
- B、 $(302)_8$
- C、 $(11000100)_2$
- D、 $(c2)_{16}$

第三题

定义变量 `int i = 0, a`，执行表达式 `a = --i` 后，`i` 和 `a` 的值分别是（ ）。

- A、-1、0
- B、0、-1
- C、-1、-1
- D、0、0

第四题

定义数组 `int a[10] = {4, 6, 1, 3, 8, 7, 2, 9, 0, 5}`，那么 `*(a + 5)` 的值是（ ）。

- A、7
- B、8
- C、2
- D、9

第五题

执行以下程序，输出的结果是（ ）。

```
int func( int x, int y, int z )
{
    if( x == 1 || y == 1 || z == 1 )
        return 1;
    if( x < y && x < z )
        return func( x, y - 1, z ) + func( x, y, z - 1 );
    if( y < x && y < z )
        return func( x - 1, y, z ) + func( x, y, z - 1 );
    return func( x - 1, y, z ) + func( x, y - 1, z );
}

int main()
{
    cout << func( 3, 3, 2 );
    return 0;
}
```

A、5

B、6

C、7

D、8

二、编程题

第一题

编程实现：求和

题目描述：

给定 n 个整数，请计算出所有大于等于 10 的整数之和。

例如： $n = 5$ ，5 个整数分别为 10、20、4、30、9，其中大于等于 10 的整数有 10、20、30，它们的和为 60（10 + 20 + 30）。

输入描述：

共两行

第一行输入一个整数 n ($1 \leq n \leq 1000$)

第二行输入 n 个整数 P_i ($1 \leq P_i \leq 100$)，整数之间以一个空格隔开

输出描述：

输出一个整数，表示所有大于等于 10 的整数之和

样例输入：

5
10 20 4 30 9

样例输出：

60

第二题

编程实现：数位和为偶数的数

提示信息：

偶数：能被 2 整除的数。

数位和：一个整数中所有数位上的数字之和。

例如：整数 123，数位和是 6 ($1 + 2 + 3$)。

题目描述：

给定一个整数 n ，请找出 1 到 n 之间（包含 1 和 n ）所有数位和为偶数的整数。

例如： $n = 15$ ，1 到 15 之间的整数为：1、2、3、4、5、6、7、8、9、10、11、12、13、14、15；

数位和依次为：1、2、3、4、5、6、7、8、9、1、2、3、4、5、6；

数位和为偶数的是：2、4、6、8、11、13、15。

输入描述：

输入一个整数 n ($2 \leq n \leq 1000$)

输出描述：

一行输出若干个整数，表示 1 到 n 之间（包含 1 和 n ）所有数位和为偶数的数，并按照从小到大的顺序依次输出，整数之间以一个空格隔开

样例输入：

15

样例输出：

2 4 6 8 11 13 15

第三题

编程实现：填涂颜色

题目描述：

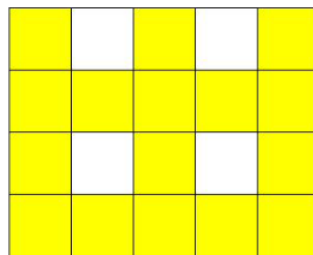
给定一个由 n 行 m 列的小方格组成的矩阵图形，接下来对该图形进行如下操作：

1、先选择其中 x 行，将其填成黄色；

2、再选择其中 y 列，将其填成黄色；

填色完成后，请统计出有多少个小方格未被填色。

例如：矩阵图形由 4 行 5 列的小方格组成，先选择第 2、4 行将其填色，再选择第 1、3、5 列将其填色。



填色完成后，有 4 个小方格未被填色。

输入描述：

共三行

第一行输入 4 个整数 n, m, x, y ，分别表示矩阵的行数和列数以及选择填色的行数和列数

($1 \leq x \leq n \leq 10000, 1 \leq y \leq m \leq 10000$)，整数之间以一个空格隔开

第二行输入 x 个不同的整数 ($1 \leq \text{整数} \leq n$)，表示被填色的行号，整数之间以一个空格隔开

第三行输入 y 个不同的整数 ($1 \leq \text{整数} \leq m$)，表示被填色的列号，整数之间以一个空格隔开

输出描述：

输出一个整数，表示填色完成后未被填色的小方格数量

4 5 2 3

2 4

1 3 5

样例输出：

4

第四题

编程实现：外观数列

提示信息：

外观数列是一个整数序列，给定该数列的第一项数据之后，从第二项开始，每一项都是对前一项数据的描述。

例如：

给定外观数列的第一项为 1；

接下来第二项是对第一项数据的描述，即“一个 1”，记作 11；

第三项是对第二项数据的描述，即“两个 1”，记作 21；

第四项是对第三项数据的描述，即“一个 2，一个 1”，记作 1211；

第五项是对第四项数据的描述，即“一个 1，一个 2，两个 1”，记作 111221；

以此类推.....

题目描述：

给定外观数列的第一项 x 以及一个整数 n ，请计算出该外观数列第 n 项的值。

例如： $x = 2, n = 3$ ，外观数列为：

2

12

1112

3112

132112

.....

外观数列第 3 项为 1112。

输入描述：

输入两个整数 x ($1 \leq x \leq 100$) 和 n ($1 \leq n \leq 30$)

输出描述：

输出一个整数，表示外观数列第 n 项的值

样例输入：

2 3

样例输出：

1112

第五题

编程实现：删除数字

题目描述：

老师在黑板上写了一个不超过 500 位的正整数 n ($1 \leq n < 10^{500}$)，要求同学们删除其中任意 k 个数字，剩余数字的顺序不变，希望得到的数最大。

例如： $n = 69134$ ， $k = 2$ ，从 69134 中删除 2 个数字，将第一位 6 和第三位 1 删除，得到的新数 934 是最大的。



输入描述：

输入两个整数 n 和 k ($1 \leq n < 10^{500}$ ， $0 \leq k < n$ 的位数)，整数之间以一个空格隔开

输出描述：

输出一个整数，表示删除 k 个数字后，得到的最大数

样例输入：

69134 2

样例输出：

934

第六题

编程实现：小松鼠的聚会

题目描述：

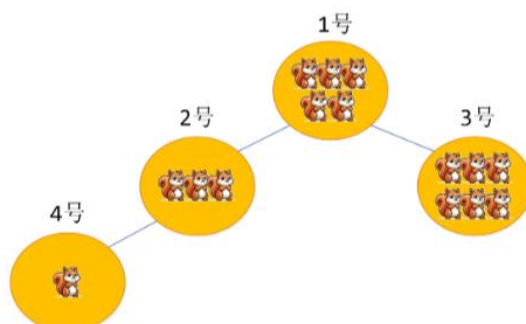
在一片树林中，有 n 个树洞，按顺序从 1 到 n 编号，每个树洞里住着至少一只松鼠。一条藤蔓连接两个树洞，共有 $n - 1$ 条藤蔓，使得任意两个树洞可以直接或间接到达。这些小松鼠经常举办聚会，当某个树洞中的小松鼠举办聚会时，它们也会邀请距离自家树洞不超过 k 条藤蔓范围的邻居们前来参加聚会。

请计算出每个树洞分别在举办聚会时，最多有多少只小松鼠参加聚会，并按照树洞编号从 1 到 n 依次输出结果。

例如： $n = 4$ ，表示有 4 个树洞，1 到 4 号树洞中居住的小松鼠的数量分别为：5、3、6、1；

共有 3 条藤蔓，每条藤蔓连接两个树洞，分别为：1 和 2、1 和 3、2 和 4；

$k = 2$ ，表示当某个树洞中的小松鼠举办聚会时，它们会邀请距离自家树洞不超过 2 条藤蔓范围的邻居们前来参加聚会；



根据上图得知：

当 1 号树洞的小松鼠举办聚会时，1、2、3、4 号树洞中的小松鼠可以参加，最多会有 15（5 + 3 + 6 + 1）只小松鼠参加；

当 2 号树洞的小松鼠举办聚会时，1、2、3、4 号树洞中的小松鼠可以参加，最多会有 15（5 + 3 + 6 + 1）只小松鼠参加；

当 3 号树洞的小松鼠举办聚会时，1、2、3 号树洞中的小松鼠可以参加，最多会有 14（5 + 3 + 6）只小松鼠参加；

当 4 号树洞的小松鼠举办聚会时，1、2、4 号树洞中的小松鼠可以参加，最多会有 9（5 + 3 + 1）只小松鼠参加；

故答案为：

15

15

14

9

输入描述：

第一行输入一个整数 n ($1 \leq n \leq 100000$)，表示树洞的数量

接下来 n 行，每行输入一个整数 c_i ($1 \leq c_i \leq 1000$)，表示每个树洞中居住的小松鼠的数量

接下来 $n - 1$ 行，每行输入两个整数 a_i, b_i ($1 \leq a_i, b_i \leq n$)，表示藤蔓连接两个树洞的编号，整数之间以一个空格隔开

最后一行输入一个整数 k ($1 \leq k \leq 20$)，表示邀请邻居的距离限制

输出描述：

共 n 行，每行输出一个整数，表示每个树洞中的小松鼠在举办聚会时，参加聚会的小松鼠的最大数量，按照树洞编号从 1 到 n 依次输出结果。

样例输入：

4

5

3

6

1

1 2

1 3

2 4

2

样例输出：

15

15

14

9